User manual for MuhRec

A. Kaestner

June 14, 2015

Contents

1. Introduction

1.1. Introduction

The development of MuhRec started when

1.2. Features

MuhRec is a reconstructor for computed tomography. It reconstructs parallel beam geometry tomographies. The software includes the following features:

- Two modes of operation; GUI and command-line.
- The GUI helps the user to set up the reconstruction and execute the processing.
- Artifact cleaning algorithms to remove ring and line artifacts.
- A guide to find center of rotation and tightest margins.
- Acquisition axis tilt correction.
- Handles image formats common at neutron imaging beamlines.
- Various options for normalization.

MuhRec3 is the next step of my reconstruction tool development. It provides more flexibility in terms of configuration. The design of the reconstruction engine has an open API that makes it possible for users to add their own processing modules. In addition the user can configure order in which the preprocessing modules are executed. Due to major changes in the architecture of the software it was not possible to maintain backwards compatibility with the configuration files from the first version of MuhRec.

1.3. Node locked software

MuhRec is free of charge and I intend to keep it so. But, since I want to have control over the distribution I decided to include a node locking routine. The node locking gives me better contact to the user community. The license key you get can only be used on the computer that generated the unlock code.

1.4. Some words about the name

The name MuhRec is derived from the sound of the cow (in German 'Muh') and Reconstructor. There are two reasons for the Muh. Firstly, the author lives in the Swiss village Muhen (which by the way has nothing to do with cows). The software was to large extent developed on the train commuting between Muhen and Paul Scherrer Institut. The development distance can hence be determined to be about 20000 km. The second reason is that the cow is the national proudness of Switzerland ...and the cow says 'Muh'. The third reason is that there are many cows surrounding PSI.

2. A First Reconstruction

2.1. Project information

When you start MuhRec, you will see the project information tab, figure (2.1). This tab allows

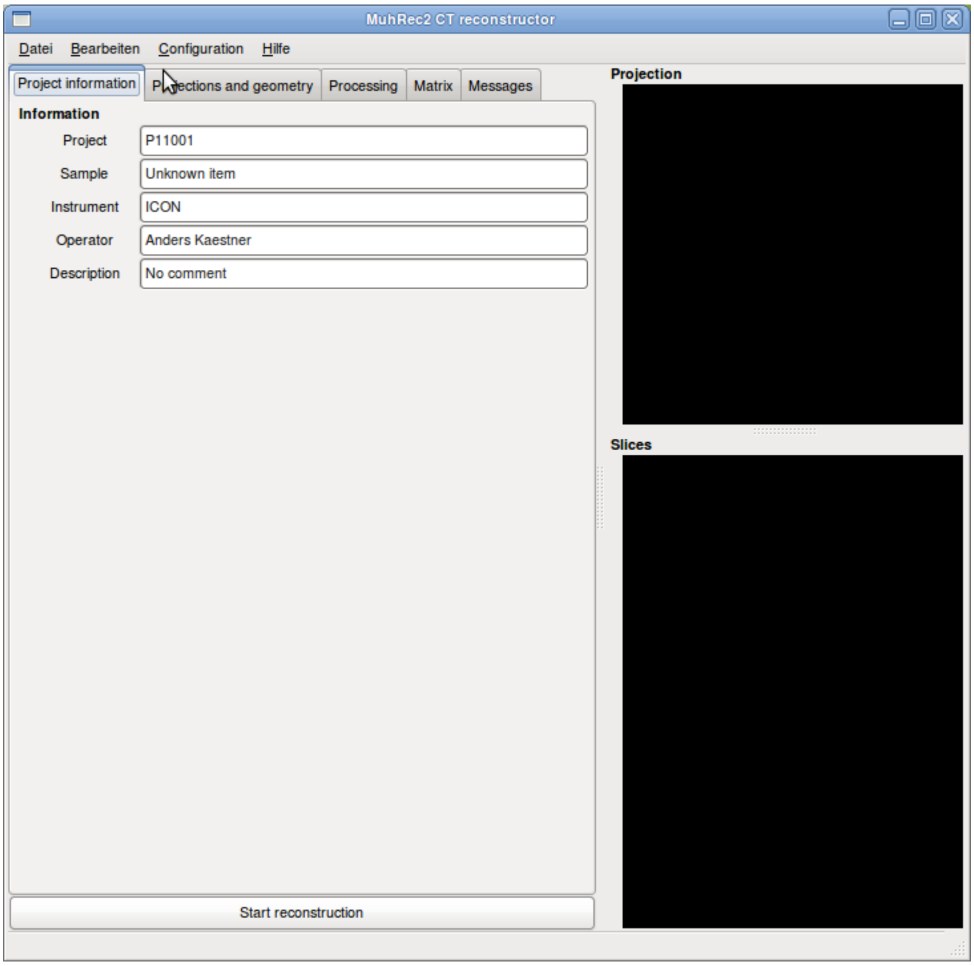


Figure 2.1.: The project information tab.

you to enter information about the experiment. The name you enter in the fields will be written to the image file headers of the reconstructed slices.

2.2. Set projection information and geometry

The next step is enter information about the projection files and the image geometry. This is done in the Projections and geometry tab, figure ().

The projection data is the input to the reconstruction. You can also provide reference images here. To enter the information you have to enter the path of the directory which contains the projection data. Then you have to enter the file mask of the projection files the general format is `basename####.ext`, where the base name can be any string. the extension is used to

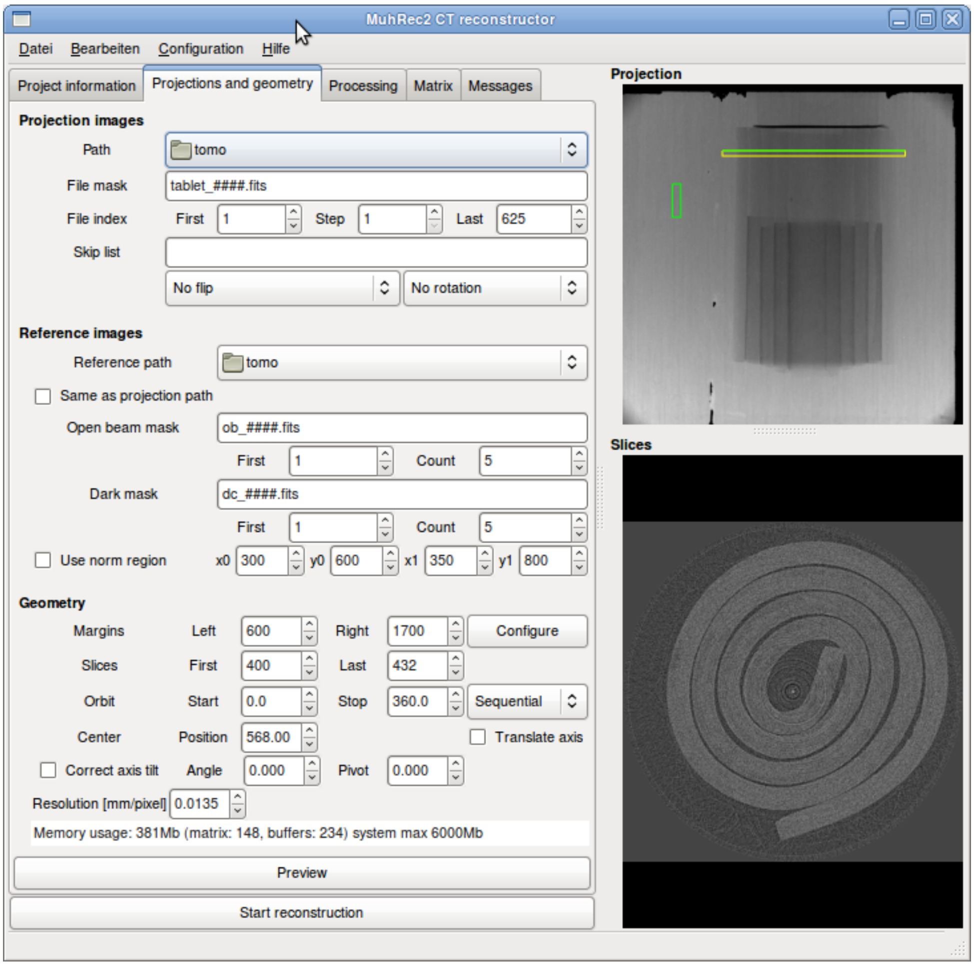


Figure 2.2.: The projection and geometry tab

determine image file type and must be correctly specified. Finally the # are place holders for the file index. The number of place holders defines how the zero padding should be used. Some examples: `proj_##.tif` will give the filenames `proj_00.tif` to `proj_99.tif`, and thereafter `proj_100.tif` etc. If you omit the # at all the indices will be inserted after the base name and before the extension without zero padding.

Once you have defined file names and path you have to set the first and last projections as well as the file stride. The stride is used to reduce the number of projections for test purposes when you want a quick reconstruction. The first index can be any positive number smaller than the last projection index.

Below the projection information you find some fields for rotation, flip, and skip projections. Theses fields are currently only place-holders for up coming features which are not even implemented.

The reference files are entered in the same manner as the projections. Both open beam and dark current images are required to located in the same directory. You also have to set the first reference image index and the number of references. These numbers are set for both open beam and dark current image separately. If you don't have any reference images just set the number of references to 0.

When all file parameters are set you can click the preview button to see the projection in the display area to the right.

2.2.1. Setting the geometry

The geometry is essential for the reconstruction. The most important is the center of rotation which has to be determined to sub-pixel accuracy. Other parameters defines the number of slices and the margins of the projections. The projection configuration dialog helps you to find the minimal margins and the center of rotation.

If your projection data was acquired with a tilted axis of rotation it will have an impact on the reconstructed data. Since this means that the center of rotation depends on the slice position, i.e. only the slice you tuned center of rotation for will be correctly reconstructed. Small tilts ($<1°$) can be corrected. The correction for larger tilt angles is still a pending feature.

Finally, you can also set the pixel size of the projections. This information is needed if you want the gray-levels of the pixels to have the unit cm^{-1}. The pixel size is also written as resolution in the slice images.

The amount of memory required for the reconstruction is estimated to give you an idea if the task is feasible for your computer. You can control the memory usage by changing the geometry of the reconstructed image, i.e. change the margins or number of slices.

The region of interest is marked in the projection preview image. The region used for dose computation is also marked in this display.

2.2.2. Saving the parameters

Once you have set all parameters it is recommended to save the parameters. You save the parameters by selecting save or save as in the file menu. Alternatively, you can also press CTRL-S which performs the same action. In a way the parameter file is a documentation of your reconstruction. It also allows you to load the parameters at later moment if you would like to do adjustments or reconstruct a new part of the samples. You can also use the parameter file for command line reconstructions and batch processing. For more details about the command line mode please read section 3.6.

The saved parameters can either be loaded into the GUI by using the File-Open menu or without GUI in the command-line option.

When you start a reconstruction the used parameters are automatically save in the file CurrentRecon.xml which is save in your home directory. This file will automatically be loaded when you start Muhrec next time.

2.3. Configure the processing chain

2.3.1. Pre-processing chain

When the geometry is configured you can switch to the processing tab, figure(2.3) On this tab, you can configure the chain of preprocessing modules. This means you can add new modules if your data requires some special treatment and you can edit the module parameters. IN the current version this can only be done in entry fields. In the future it is planed to add wizards to guide the configuration of the more complex modules.

2.3.2. The back-projector

It is also possible to change back-projector and configure its parameters. The most important parameter is the buffer size. This parameter has a direct effect on the memory consumption during the reconstruction. On a 32-bit system it may be needed to change the buffer to a smaller value when wide projections are reconstructed.

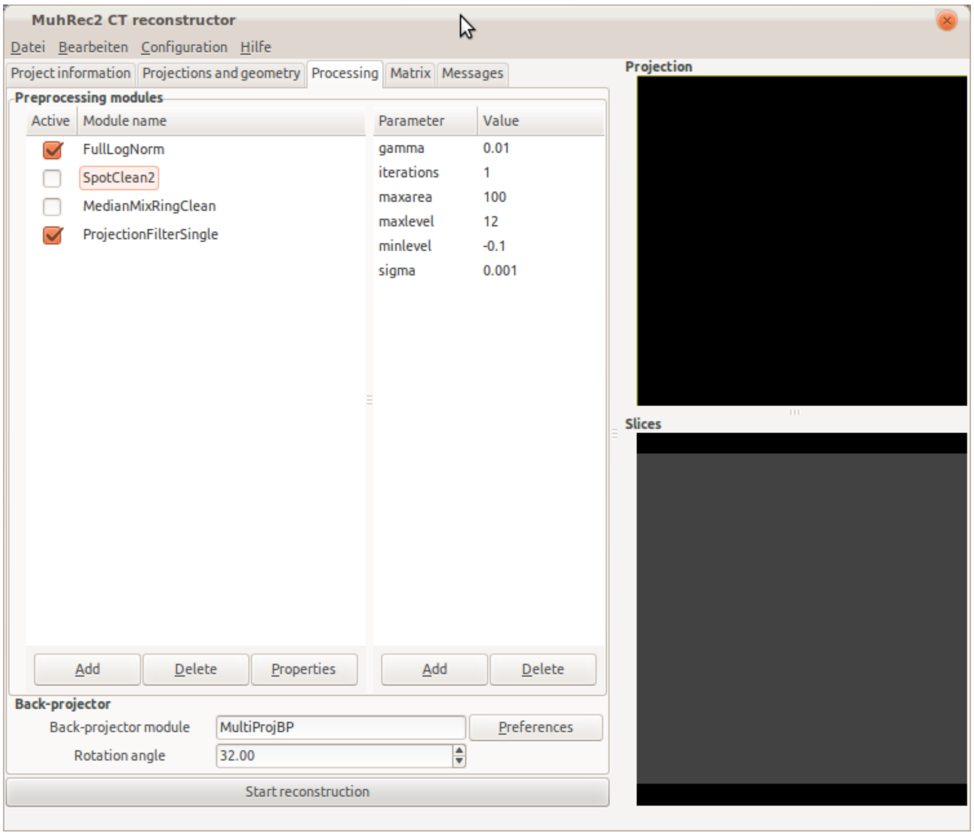


Figure 2.3.: The process chain configuration tab.

2.4. Reconstruction

2.4.1. Start the processing

Once the parameters for the reconstruction have been set you can start the reconstruction by pressing the 'Start Reconstruction' button. There are however two modes to consider; for small data sets you can do the reconstruction in interactive mode. This means that you after the finished reconstruction can adjust the gray levels and select destination folder, file name, and file format. When the selected matrix is to large for the available memory a dialog that gives you the options to reconstruct directly to disk with the setting on the matrix tab or to cancel the started reconstruction. If you select the direct streaming option for the reconstruction there will be no output in the slice display.

Often it is recommended to make a small test reconstruction to check the quality of the selected parameters. Select between 16 and 64 slices in a relevant region of the sample and reconstruct them. It may take some iterations until you have everything as you want it. When you are satisfied with the result you can reconstruct the whole data set in a single run. This may take some time.

2.4.2. Finalize the data

When the reconstructor finished processing the data you are automatically directed to the 'Matrix'-tab, figure 2.4.

In this mode you see a histogram of the reconstructed data. The vertical lines in the histogram shows the interval used to display the reconstructed slice in the viewer. These levels will also be used to in the conversion to integer formats. You can change the gray level interval in the entry fields on the top.

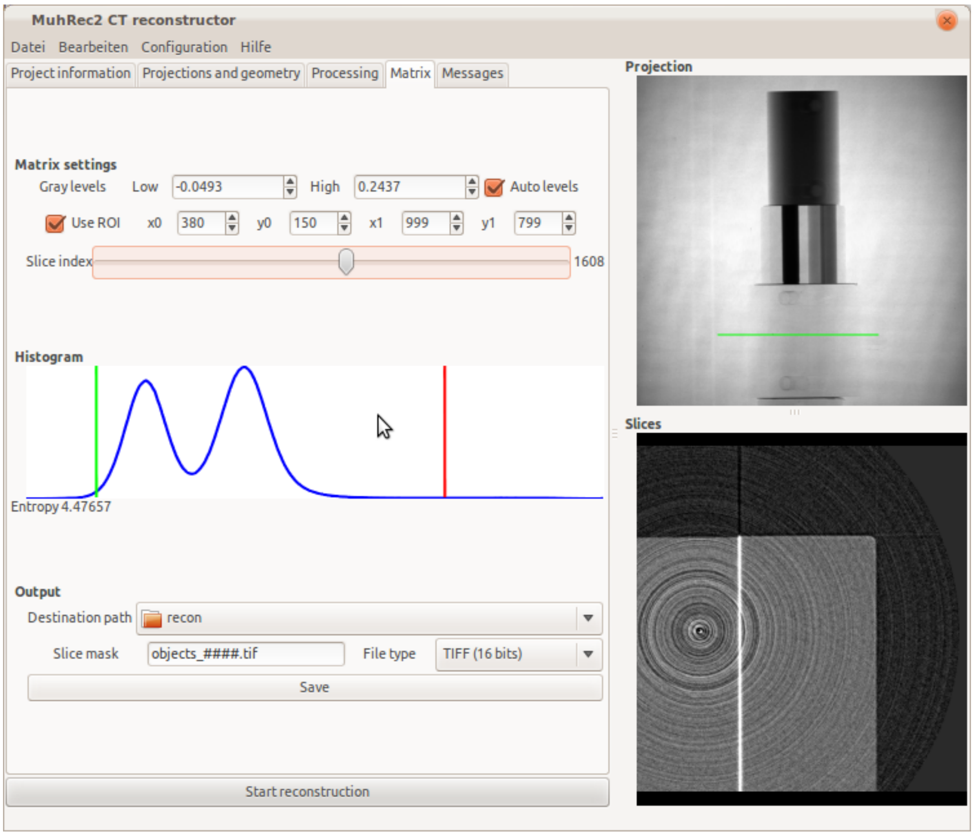


Figure 2.4.: The matrix display tab.

A slider can be used to browse through the slices. By sliding through the slices you can inspect the slices to check that the gray interval is correctly set. You can also see if there are any artifacts in your data.

Once you are satisfied the image settings and reconstruction quality you can save the slices. To do this you have to select the destination folder and enter the file mask for the slices. The mask has the same format as the projection data i.e. `base####.ext`. The extension must be entered manually as it is not adapted to the selected file type. The last selection to make is the file format which can be either tiff with different bit depths or matlab binary.

2.5. The geometry list

It is possible to store the current configuration in the geometry list using the application menu item Geometry. This can be used to test the effect of different settings without losing the original settings. Each parameter set will be stored with the central slice to make comparison easier. If you have reconstructed several slices on different places you can use the entries of the list to calculate the axis tilt.

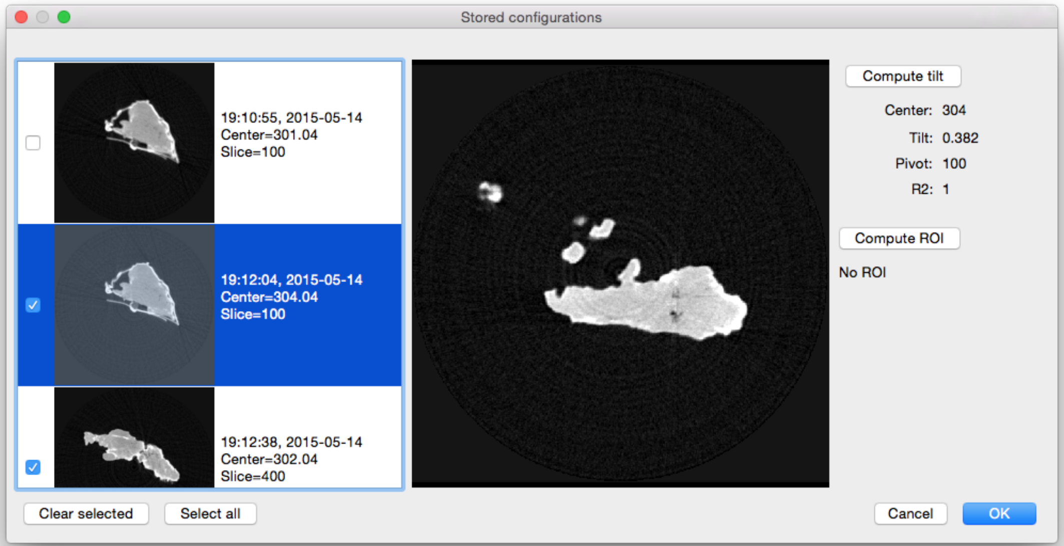


Figure 2.5.: The Geometry list dialog.

3. Detailed descriptions

3.1. Processing modules

The processing module play a central roll in the reconstruction. They are used to manipulated the projection data prior to the back projection task. In the following sections each module is described. The order of the modules can be arbitrarily configured. In some cases it makes sense to maintain a certain order among the modules. This reordering can be done in the GUI by dragging the modules in the pre-processing module list. Unfortunately, there is a bug in the GUI package that may cause the application to crash. The reordering can also be made in the config file since the modules are executed in the order they are entered.

The modules can be configured by clicking on a module in the module list. Then, a list of parameters for the module is shown. Click on the parameter value you want to change and enter the new value. Don't forget to press enter to confirm the new entry. A module can also be temporarily disabled by unchecking the active check-box. Then it remains in the configuration but does not contribute to the processing chain. This option is useful if you want to speed up the processing a little. It can also be used to demonstrate effect before and after adding the module.

3.1.1. FullLogNorm

In most cases a complete projection data set also contains open beam and dark current images. The location and names of the images are entered separately under the 'Projections'-tab. Here, you can also set the number of available reference images. The full normalization operation is computed using

$$p = -\log\left(\frac{D_0}{D} \cdot \frac{I - I_{DC}}{I_{OB} - I_{DC}}\right) \tag{3.1}$$

This equation includes correction for variations in the neutron dose.

This module computes the flat field correction of the projections. The correction is followed by the negative logarithm. The module uses the reference images entered in the projection information tab.

uselut Selects if the computation should be supported by a LUT to gain some speed.
Values:true/false.

usenormregion Selects if the norm region should be used for dose correction.
Values:true/false.

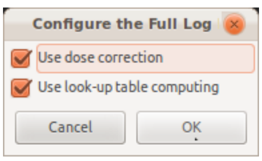


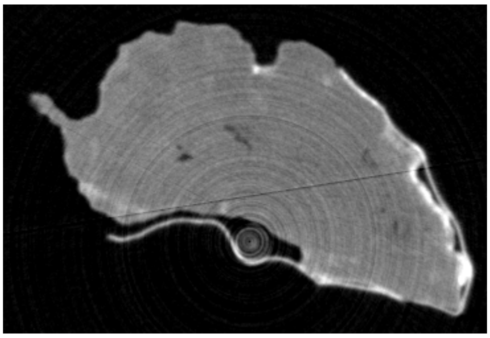
Figure 3.1.: Configuration dialog for the log norm module.

Object file: libStdPreprocModules.so

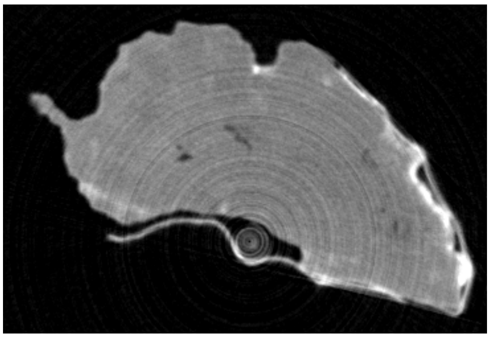
3.1.2. SpotClean2

Spots are the most frequent outlier artifacts encountered in neutron imaging. These appear as line artifacts in the reconstructed data. There is a large benefit in using the spot-cleaning algorithm since this increases the signal to noise ratio in the matrix significantly.

Line artifact cleaning is parameterized by two parameters; the threshold level and a fuzziness parameter around the threshold. Mostly, the fuzzy width parameter is set to about 10% of the threshold.



Original



Cleaned

Figure 3.2.: The effect of spot cleaning.

The spot cleaning is parameterized by two parameters; the threshold and the mixing fuzziness. The threshold defines the intensity of the pixels to correct. All pixels with values exceeding the threshold will be corrected. The second parameter is used for interpolation between the estimated correction values and the original image.

$$p_{corrected} = (1-w)\,p_{original} + w\,p_{estimate}$$

the weight function is a sigmoid centered about the threshold level. This means that some pixels below the threshold will also be involved in the correction.

The spot cleaning configuration wizard helps you to find the correct levels of the two parameters. A threshold function plot in the cumulative histogram shows in which interval the correction will work. When you execute the correction the detection map and the the corrected image will be shown. The amount of corrected pixels will also be displayed. This wizard will only show you the region of interest you marked in the projection geometry setting. Therefore, you may want to increase the number of slices a while for a better overview during the tuning process of the spot cleaning.

gamma This is a threshold value setting an allowed variance maximum that discriminates local image variations from actual outliers.
Typical value: 0.03

iterations This is currently an unused parameter that would iterate the cleaning procedure N times.
Typical value: 1

maxlevel Data clamping parameter, values greater than this level are set to the defined level.
Default value: 12

minlevel Data clamping parameter, values less than this level are set to the defined level.
Typical value: 0

sigma A width parameter to allow smooth mixing of the values around the threshold.
Default value: 0.005

Object file: libStdPreprocModules.so

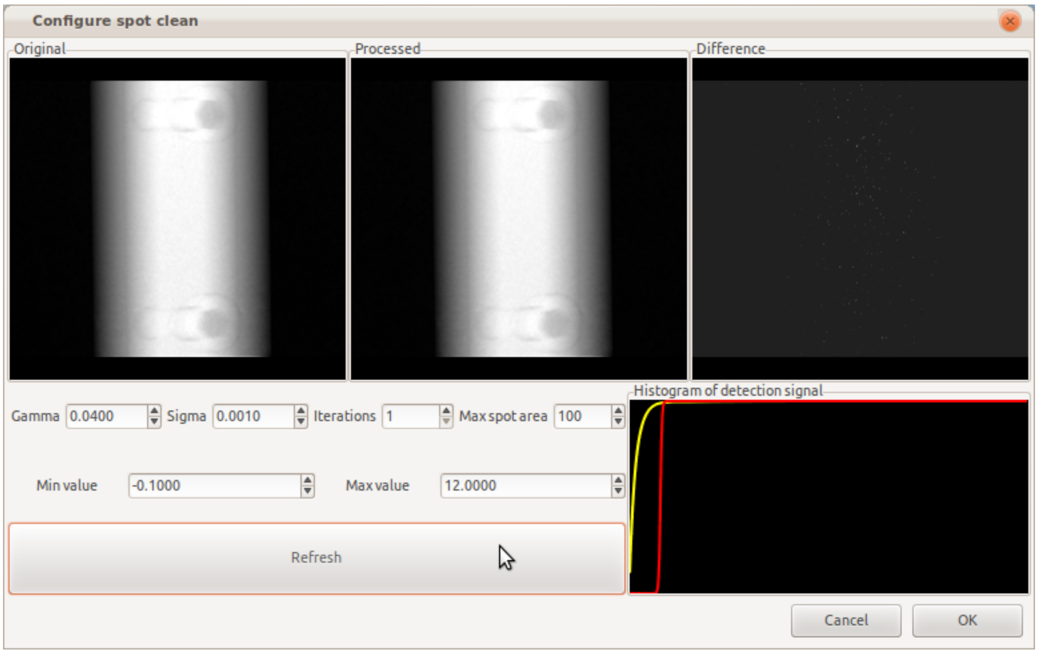


Figure 3.3.: The spot clean 2 configuration dialog.

3.1.3. MedianMixRingClean

Ring artifacts are with line artifacts the most common artifact in a CT slice. This is a basic method for correcting ring artifacts. It identifies outliers in the average sinogram and corrects them by the local median.

threshold Sets a threshold on the local variance that identifies an outlier that would cause a ring.
Default value: 0.01

width Sets a smooth weighting interval for the replacement. Typically this value is about 10% of the threshold.
Default value: 0.001

Object file: libStdPreprocModules.so

3.1.4. ProjectionFilterSingle

This is a central module for the filtered back-projection algorithm. It applies a ramp filter combined with a apodization window line-wise on the projection data.

cutoff The cut-off frequency can be used to improve the signal to noise ratio of the reconstruction. As it has a low-pass effect it will smooth the edges in the image. It can take values between 0 and 0.5.
Default value: 0.5

filtertype This parameter controls the shape of the apodization filter. There are several window shapes implemented: ramlak, shepplogan,hanning,hamming, and butterworth.
Default value: hamming

order This parameter is only relevant for the Butterworth filter where it defines the exponent of the filter and thus controls the filter shape.
Default value: 0

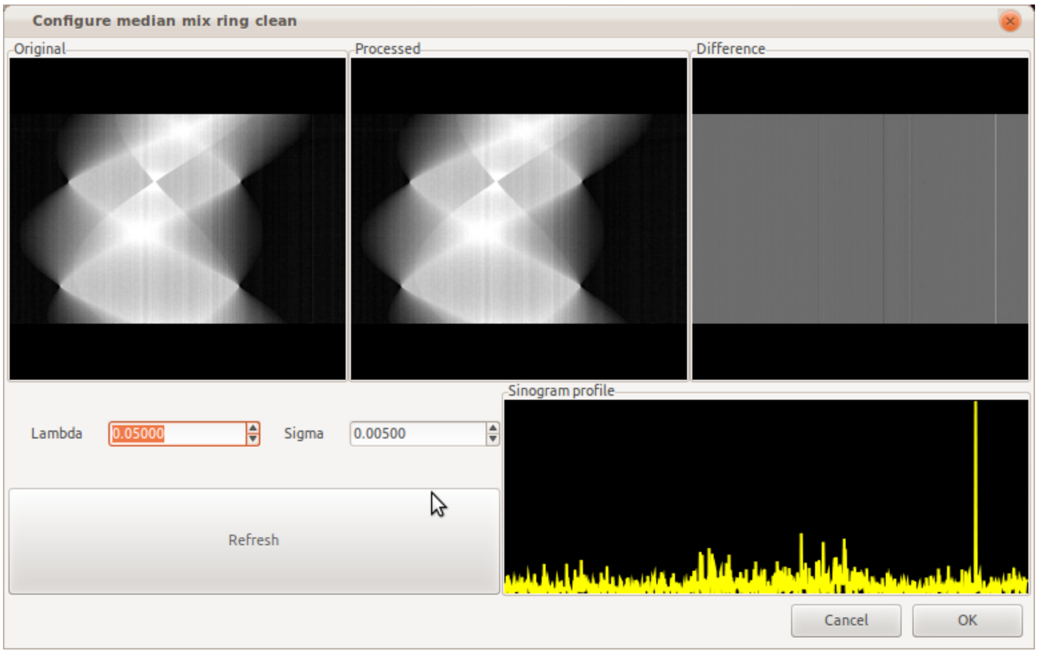


Figure 3.4.: The median mix ring clean configuration dialog.

usebias The ramp filter cancels the DC term of the spectrum. The effect is a bias in the reconstruction. By activating this parameter a small value is added to the DC component. Default value: true

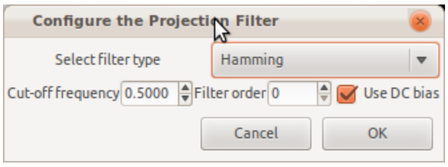


Figure 3.5.: The projection filter configuration dialog.

Object file: libStdPreprocModules.so

3.1.5. ISSfilter

This is an experimental module used to smooth the projection data. The module uses an inverse scale space filter [?] to filter the individual projections. It seems to improve the SNR with maintained edge sharpness. The filter has some problems with the numerical stability, at least it is difficult to tune. There will be a tuning wizard in the future.

N Number of solution iterations.
Default value 10

alpha 0.25

lambda 1

tau 0.02

Object file: libStdPreprocModules.so

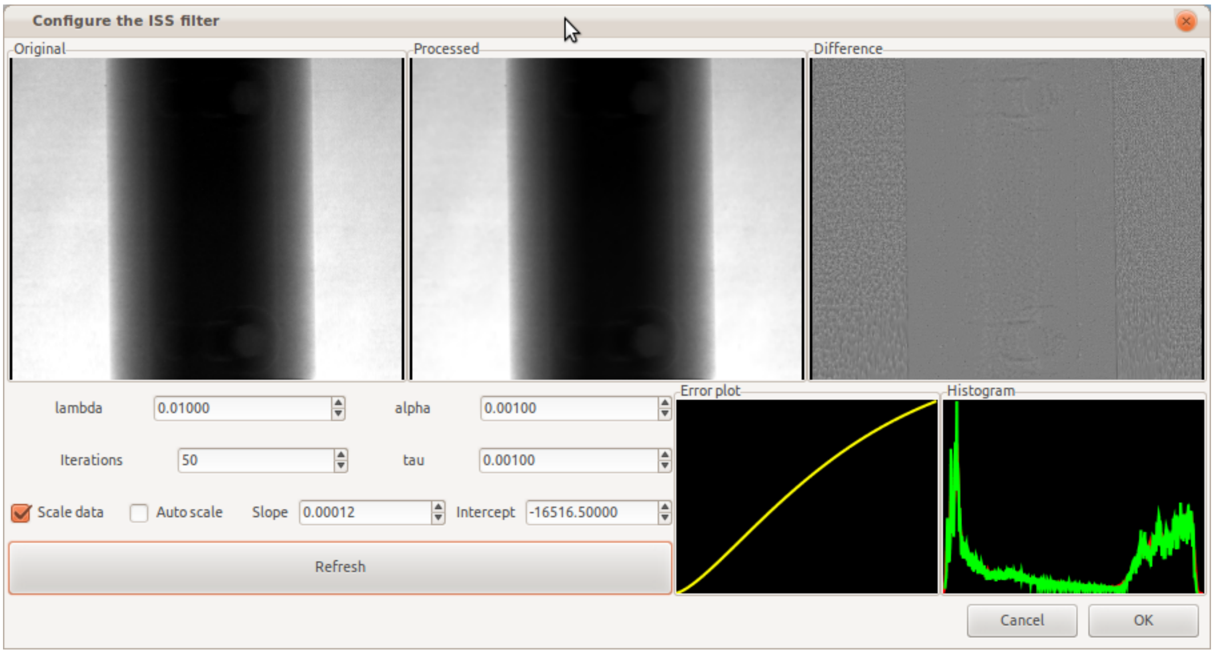
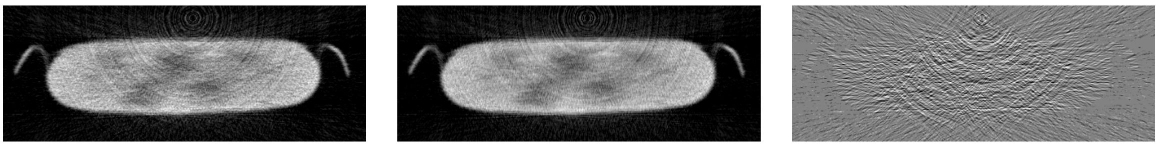


Figure 3.6.: The projection filter configuration dialog.

3.1.6. AdaptiveFilter

Samples with high aspect ratio between width and thickness have an orientation dependent signal to noise ratio[?]. This filter adaptively applies more smoothing to regions with low transmission.



(a) (b) (c)

Figure 3.7.: The effect of the module for adaptive filtering. The unfiltered image (a), reconstruction with the adaptive filter activated (b), and (c) is difference image between the images (a) and (b).

lambda Threshold at with the smoothing takes place. The value of lambda makes sense in the interval [0.0, 1.0].
Default value: 0.1

sigma Defines the width of the mixing interval. This value should mostly be less than 0.1.
Default value: 0.05

size Width of the smoothing filter kernel.
Default value: 5

Object file: libStdPreprocModules.so

3.1.7. BasicRingClean

The basic ring clean subtracts a high-pass filtered average sinogram from all sinogram. This a rather crude way of ring correction but it will cancel some of the rings.
Object file: libStdPreprocModules.so

3.1.8. DataScaler

The data scaler makes a linear scaling of the data $k\,x+m$. This can be useful if have preprocessed images that are save in a 16-bit data format.
Object file: libStdPreprocModules.so

offset The offset (m) of the straight line.
Default value: 0.0

slope The slope (k) of the straight line.
Default value: 1.0

3.1.9. GeneralFilter

Object file: libStdPreprocModules.so

size 3

type box

3.1.10. PolynomialCorrection

This module applies a polynomial to each pixel value. This is used to correct for beam hardening effects in the reconstructed image. Object file: libStdPreprocModules.so

Coefficents The coefficients of the terms of the polynomial starting with a_0 i.e. there must be PolynomialDegree+1 values.
Default value: 0.0 1.0

PolynomialDegree Sets the polynomial degree, the module can handle up to the 8th degree.
Default value: 1

3.1.11. SpotRingClean

Ring cleaning by spot cleaning is an experimental module that is not finished. It cleans some of the rings but the performance is still not satisfying.
Object file: libStdPreprocModules.so

gamma Threshold level in the detection image.
Default value: 0.01

iterations Number of times the filter should be repeated.
Default value: 1

sigma Width of the mixing interval.
Default value: 0.001

3.1.12. TranslatedProjectionWeighting

This module weights the projections near the center of rotation to provide a smooth stitching of the projection data when you reconstruct data that is acquired with the center near the edge. This is a method to increase the FOV at maintained resolution.
Object file: libStdPreprocModules.so

weightfunction sigmoid

width 5

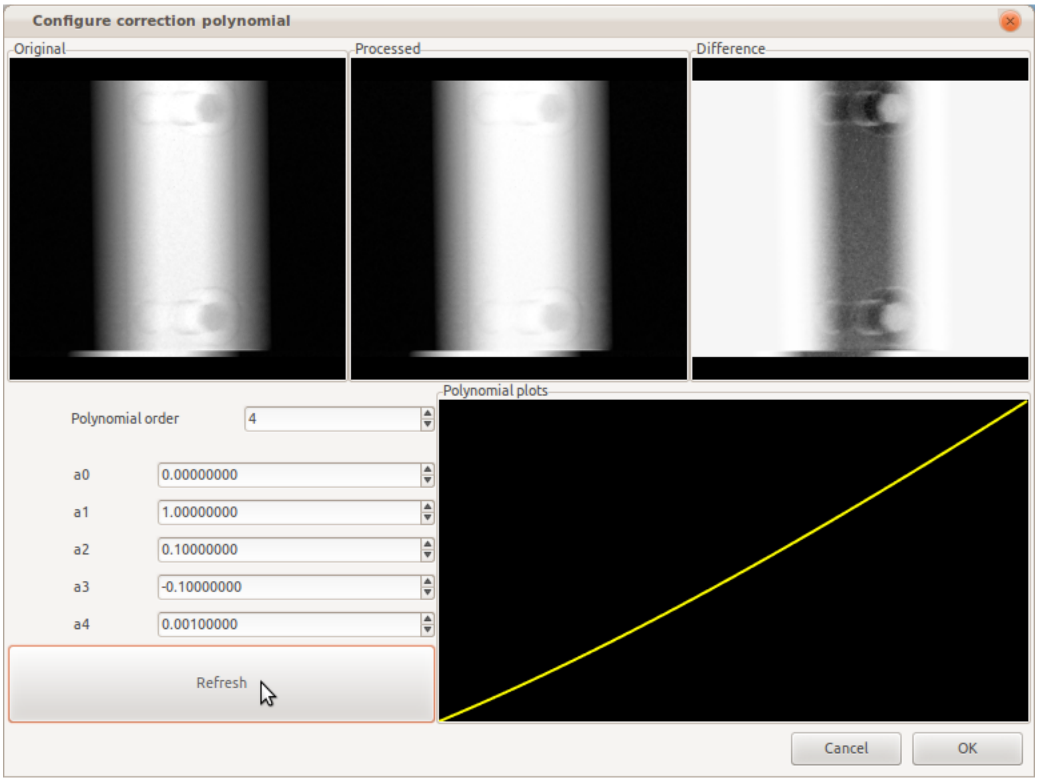


Figure 3.8.: The projection filter configuration dialog.

3.1.13. WaveletRingClean

This ring cleaning module works in the sinogram domain. It does the cleaning using a combination of wavelet and fourier transforms[?]. The current version introduces new artifacts as a consequence of bad edge processing in the wavelet transform. I.e. this module is still under development. The filter bank could also gain of the addition of more and longer filter descriptions. Object file: libStdPreprocModules.so

decnum Number of decomposition levels.
Default value: 5

sigma Width of the Gaussian that forms the stop band.
Default value 0.1

wname The name of the wavelet base.
Default value: daub25

3.1.14. CountNANs

This module is mainly for debugging use. It counts the number of pixels set to NaN. This can be helpful to identify which module causes the numeric errors in your processing chain.
Object file: libInspectorModules.so

3.1.15. ProjectionInspector

This module is not fully implemented but is intended to open a display window that shows the projections in their current condition.
Object file: libInspectorModules.so

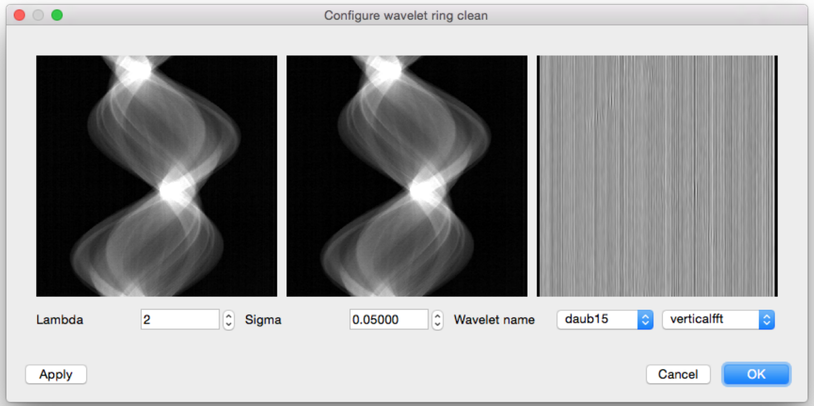


Figure 3.9.: Dialog to tune the ring correction using wavelet ringclean. Left panel is original, mid shows the processed sinogram, and the panel to the right shows the difference.

3.1.16. SaveProjections

This module save the projection data in the current state of processing as a set of 32-bit tif images.

Object file: libInspectorModules.so

path Path to the destination folder

filemask File mask of the written files. It shall always contain a block of #'s as placeholder for the file index. Default value:`./projections_####.tif`

imagetype Selects the file type of the projection data. It can be either projections or sinograms. Default value: projections.

filetype Selects the precision of the data in the written files. It can be 8- or 16- bit unsigned or 32 bit floating point.
Default value: 32bit floating point.

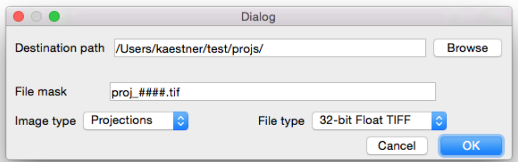


Figure 3.10.: Dialog to set the values of the SaveProjections module.

3.2. The Viewer

The viewer area on the right hand side of the application main window is used to display projections and reconstructed slices. If move mouse pointer to the viewer and wait shortly a tool tip message will appear. It tells you the position and intensity on position you point with the mouse. This is useful to determine threshold levels guiding algorithms like the margin detector.

The contrast and brightness can be adjusted by dragging using the right mouse button. If you pause while the right mouse button is pressed, a tool tip shows the current brightness and contrast.

A level dialog appear if you press ”l’ when the viewer is active. You can also mark regions in the viewer and press the Get ROI button related to the viewer.

3.3. The projection data

MuhRec supports input projection data in the formats: TIFF, FITS, and Matlab binary. Normally the filenames of the projections are sequentially ordered and each file has a base name and an index number. To enter the file name in the projection entry field you should use the format `base_####.ext`. The number of # tells MuhRec how many zeros to use in the formatting. The #'s can be placed anywhere in the name string. The extension indicates which file format to use.

Below the file name field the file interval can be entered with first and last file index. These indices will both both included. In addition there is also an entry to set the increment. This increment can be used to decrease the number of projection in test reconstructions. This saves time, but do not forget to repeat the reconstruction with all projections before you start the final reconstruction.

An alternative way to enter the projection data with a text file. The advantage of this approach is that you can use any filename and for each projection and you can specify the acquisition angle for each projection. The format of the file is

```
angle0 <tab> fileA.ext
angle1 <tab> fileB.ext
.
.
.
angleN <tab> fileXYZ.ext
```

The you use a text file to specify the projection data. You can use the file index numbers to select intervals in the file list. The first line in the file has index 1. Stepping does not work.

Once you have entered all projection information you can press the 'Preview'-button and the first projection will appear in the viewer. The displayed projection is the raw projection.

3.4. Spatio-temporal tomography

For samples which change over time it is not possible to

3.5. Tilt correction

When the turn table and detector are not aligned there will be an error in the reconstructed data. For small deviations this error can be corrected by adjusting the center of rotation for each slice. In MuhRec you have two parameters for the tilt correction. The first is the axis tilt angle and the other set the pivot point relative to the image.

The pivot is used for long samples where the turn table is located far away from the detector. For a single data set this parameter is less important than when you have several scans that you want to merge after the completed reconstruction. It is not possible to correct for both detector and pivot tilt simultaneously in the current version. Figure 3.11 shows the two different detector rotations.

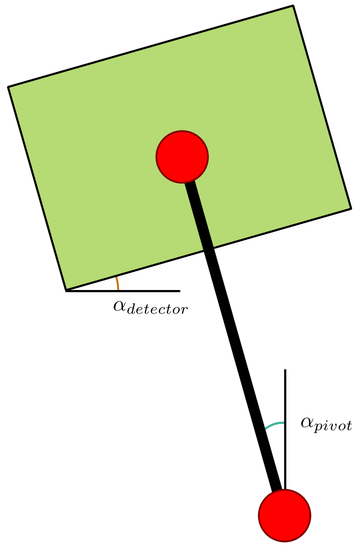


Figure 3.11.: Possible acquisition axis tilts.

3.6. Reconstructing from the command line

In addition to the GUI you can also start MuhRec from the command line. This can done by typing

```
muhrec3 -f parameters.xml
```

The file parameters.xml contains all settings needed for the reconstruction. This is also the same type of files written by GUI. In appendix B an example of the parameter file is given.

The command line option is useful for batch reconstructions. The parameter files for all reconstruction jobs can be prepared with the GUI. In some cases there are only a few parameters that change between the data sets. In this case you can add these parameters on the command line using the format:

```
muhrec3 -f parameters.xml block1:parameter1=value1 block2:parameter2=value2
```

The `block` is the xml block in the parameter file (userinformation, system, projections, matrix) and `parameter` is the parameter you want to modify with `value`. Please note that each parameter argument must be a single string each space will be interpreted as delimiter for a new argument. Therefore if you want to change an array parameter you have to use "..." to enclose the argument. Eg.

```
muhrec3 -f parameters.xml "projections:roi=10 10 100 100"
```

Muhrec use less memory when it is operated in CLI mode.

Invoking Muhrec from a script

Sometimes it is convenient to use a script to run muhrec. Here is an example using python

```
#!/usr/bin/env python

print("Processing some frames.")

projpath="/Volumes/DataDisk/P20140142"
destpath=projpath+"/04_evaluation/20141126"
```

```
muhrec="/Users/kaestner/Applications/muhrec3.app/Contents/MacOS/muhrec3"
cfgpath=projpath+"/04_evaluation/20141126/recon_roots.xml"

from subprocess import call
from math import fmod
firstproj=1;
firstframe=0
lastframe=3
for i in range(firstframe,lastframe) :
# select projection sub set
firstindex="projections:firstindex="+str(firstproj+i*180)
lastindex="projections:lastindex="+str(firstproj+(i+1)*180)
# set file mask for the slices
matrixname="matrix:matrixname=frame_"+("%04d" % i)+"-slice_####.tif"
# adjust the reconstruction angles to alternating between 0-180 and 180-360
angle=fmod(i,2)*180
scanarc="projections:scanarc="+str(angle)+" "+str(angle+180)
# call the reconstruction
call([muhrec, "-f", cfgpath, firstindex, lastindex, matrixname, scanarc])
```

This script assumes that you have a well configured Recon.xml file and that the data is acquired as a long sequence, e.g. a time series of CTs.

4. Continued development

MuhRec is continuously developing to adapt to new challenges at the neutron imaging beamlines at Paul Scherrer Institut. Unfortunately, it is not only new development but also bug fixing. To fix bug the input from users is important, so I kindly ask you to report problems when the software misbehaves.

For the continued development it also important with feedback regarding improvements of existing and new features. So, if you have any comments that you would like to share I appreciate if you send them to me. I cannot promise that I will implement everything but I will at least consider it.

Bibliography

A. License

Definitions: The software refers to any revision of MuhRec. The author Anders Kaestner is referred to as the author. Any person who installed and intends to use or uses the software is referred to as the user.

- The software (source code, binaries, and any supporting material) is the property of the author.
- The user is allowed to install and use the software free of charge.
- It is not allowed to decompile and/or modify the files in the software distribution.
- The software is delivered as is. The author gratefully receives bug reports and feature requests but can not promise any updates. In case of updates the user will be notified.
- The author takes no responsibility for the use of the software and the accuracy of the results.
- The user is *strongly discouraged* from using the software for medical applications.
- If you use the results in a publication. The following publication should cited: Anders P. Kaestner, *MuhRec – A new tomography reconstructor*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment Volume 651, Issue 1, 21 September 2011, Pages 156-160, doi:10.1016/j.nima.2011.01.129

B. Parameter file format

The parameter file is formatted with xml and is divided into subsections. Here the default parameter file is shown:

```
<reconstructor>
    <userinformation>
        <operator>Anders Kaestner</operator>
        <instrument>ICON</instrument>
        <projectnumber>P11048</projectnumber>
        <sample>Curse tablet</sample>
        <comment>No comment</comment>
    </userinformation>

    <system>
        <memory>6000</memory>
        <loglevel>message</loglevel>
    </system>

    <projections>
        <dims>2048 2048</dims>
        <resolution>0.0135 0.0135</resolution>
        <firstindex>1</firstindex>
        <lastindex>625</lastindex>
        <projectionstep>1</projectionstep>
        <repeatline>false</repeatline>
        <scantype>sequential</scantype>
        <imagetype>projections</imagetype>
        <center>568</center>
        <translation>false</translation>
        <tiltangle>0</tiltangle>
        <tiltpivot>0</tiltpivot>
        <correcttilt>false</correcttilt>
        <filemask>tablet_####.fits</filemask>
        <path>/home/data/P11048_tablet/tomo/</path>
        <referencepath>/home/data/P11048_tablet/tomo/</referencepath>
        <obfilemask>ob_####.fits</obfilemask>
        <obfirstindex>1</obfirstindex>
        <obcount>5</obcount>
        <dcfilemask>dc_####.fits</dcfilemask>
        <dcfirstindex>1</dcfirstindex>
        <dccount>5</dccount>
        <roi>600 400 1700 432</roi>
        <doseroi>300 600 350 800</doseroi>
        <scanarc>0 360</scanarc>
    </projections>
```

```
<matrix>
    <dims>1100 1100 32</dims>
    <rotation>0</rotation>
    <serialize>false</serialize>
    <path>/home/kaestner/work/svn/</path>
    <matrixname>tablet_####.tif</matrixname>
    <filetype>TIFF16bits</filetype>
    <firstindex>0</firstindex>
    <grayinterval>-1 0.728486</grayinterval>
</matrix>

<processchain>
    <preprocessing>
        <module>
            <modulename>FullLogNorm</modulename>
            <sharedobject>libStdPreprocModules.so</sharedobject>
            <active>true</active>
            <parameters>
                <uselut>false</uselut>
                <usenormregion>true</usenormregion>
            </parameters>
        </module>
        <module>
            <modulename>SpotClean2</modulename>
            <sharedobject>libStdPreprocModules.so</sharedobject>
            <active>true</active>
            <parameters>
                <gamma>0.03</gamma>
                <iterations>1</iterations>
                <maxlevel>12</maxlevel>
                <minlevel>0.0</minlevel>
                <sigma>0.005</sigma>
            </parameters>
        </module>
        <module>
            <modulename>MedianMixRingClean</modulename>
            <sharedobject>libStdPreprocModules.so</sharedobject>
            <active>true</active>
            <parameters>
                <threshold>0.01</threshold>
                <width>0.001</width>
            </parameters>
        </module>
        <module>
            <modulename>ProjectionFilterSingle</modulename>
            <sharedobject>libStdPreprocModules.so</sharedobject>
            <active>true</active>
            <parameters>
                <cutoff>0.5</cutoff>
                <filtertype>hamming</filtertype>
                <order>0</order>
```

```
                <usebias>true</usebias>
            </parameters>
        </module>
    </preprocessing>

    <backprojector>
        <module>
            <modulename>MultiProjBP</modulename>
            <sharedobject>libStdBackProjectors.so</sharedobject>
            <active>true</active>
            <parameters>
                <ProjectionBufferSize>16</ProjectionBufferSize>
                <SliceBlock>64</SliceBlock>
                <SubVolume>1</SubVolume>
            </parameters>
        </module>
    </backprojector>

</processchain>

</reconstructor>
```

Most of the parameters can be set with GUI. There are however some exceptions.

- `<memory>` amount of memory available for reconstruction, unit mb.
- `<projectionbuffersize>` number of projections to process simultaneously.
- `<sliceblock>` max number of slices to process in one processing block.

The buffer parameters have direct impact on the reconstruction time.

C. Intended new features

1. Flip, and 90° rotations.
2. True rotation of the projections.
3. Center estimation by reconstruction.
4. Add a module for QNI for the projections. QNI is a method to reduce the impact of scattering on the data. It often makes it possible to quantify the water content in a sample.
5. Your suggestions.

D. Known bugs and limitations...

D.1. Limitations

1. The back-projection core only works on a single thread.
2. Center of rotation can not be found for translated projection data.

D.2. Bugs

- The reconstructor may crash when the number of slices is not a factor of 4.
- The translated center reconstruction does not reconstruct correctly near the center of rotation. This is especially noticeable when tilt correction is activated
- First and last slice in a block appears to have different intensity. Not verified.
- The module selection list in add module is not cleared when a new object file is opened.